

Assignment #6: Recursion and DP

Updated 2201 GMT+8 Oct 29, 2024

2024 fall, Compiled by 昂奕, 化学与分子工程学院

说明:

- 1) 请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora <https://typoraio.cn>，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。
- 3) 提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、“作业评论”区有上传的md或者doc附件。
- 4) 如果不能在截止前提交作业，请写明原因。

1. 题目

sy119: 汉诺塔

recursion, <https://sunnywhy.com/sfbj/4/3/119>

思路: 先把 (n-1) 个盘子移到B, 再把1个盘子移到C, 最后把 (n-1) 个盘子移到C。

代码:

```
n=int(input())
i=0
def f(n,p1,p2):
    global i
    if n==1:
        i+=1
        return f'{p1}->{p2}'
    else:
        p2_=list({'A','B','C'}-{p1,p2})[0]
        return f(n-1,p1,p2_)+'\n'+f(1,p1,p2)+'\n'+f(n-1,p2_,p2)
ans=f(n,'A','C')
print(i)
print(ans)
```

完美通过

100% 数据通过测试

运行时长: 0 ms

语言: Python

```
1  n=int(input())
2  i=0
3  def f(n,p1,p2):
4      global i
5      if n==1:
6          i+=1
7          return f'{p1}->{p2}'
8      else:
9          p2_=list({'A','B','C'}-{p1,p2})[0]
10         return f(n-1,p1,p2_)+'\n'+f(1,p1,p2)+'\n'+
11 ans=f(n,'A','C')
12 print(i)
13 print(ans)
```

sy132: 全排列I

recursion, <https://sunnywhy.com/sfbj/4/3/132>

思路：假设有n位，先从其中去除1个数，排剩下（n-1）个数，再把取出的数加到排好的序列每一项前面

代码：

```
def A(n):
    if len(n) == 1:
        return [n]
    else:
        ans=[]
        for i in range(len(n)):
            tmp=A(n[:i]+n[i+1:])
            for j in tmp:
                ans.append([n[i]]+j)
        return ans
x=int(input())
n=list(range(1,x+1))
```

```
for i in A(n):  
    print(*i, sep=' ')
```

代码运行截图 == (至少包含有"Accepted") ==

完美通过

100% 数据通过测试

运行时长: 0 ms

语言: Python

```
1  def A(n):  
2      if len(n) == 1:  
3          return [n]  
4      else:  
5          ans=[]  
6          for i in range(len(n)):  
7              tmp=A(n[:i]+n[i+1:])  
8              for j in tmp:  
9                  ans.append([n[i]]+j)  
10         return ans  
11  x=int(input())  
12  n=list(range(1,x+1))  
13  for i in A(n):  
14      print(*i, sep=' ')
```

02945: 拦截导弹

dp, <http://cs101.openjudge.cn/2024fallroutine/02945>

思路:

状态转移方程:

```
f(i,j)=  
    1+f(i-1,i),打  
    f(i-1,j),不打
```

i: 剩余导弹数

j: 最大高度是第几个导弹的高度

代码:

```

k=int(input())
h=list(map(int,input().split()))+[-1]
h.reverse()
h.append(1e15)
mat=[-1]+[0 for i in range(k-j)] for j in range(k)]
for _ in range(k+1,1,-1):
    mat[1][k+1-]=int(h[_]>=h[1])
for i in range(2,k+1):#行, h[i]是现在需要打不打的那个的高度
    for j in range(k+1,i,-1):#列, h[j]是最大能打高度.
        if h[j]>=h[i]:
            mat[i][k+1-j]=max(1+mat[i-1][k+1-(i)],mat[i-1][k+1-j])
        else:
            mat[i][k+1-j]=mat[i-1][k+1-j]
print(mat[-1][-1])

```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```

k=int(input())
h=list(map(int,input().split()))+[-1]
h.reverse()
h.append(1e15)
#print(h)
mat=[-1]+[0 for i in range(k-j)] for j in range(k)]
for _ in range(k+1,1,-1):
    mat[1][k+1-]=int(h[_]>=h[1])
for i in range(2,k+1):#行, h[i]是现在需要打不打的那个的高度
    for j in range(k+1,i,-1):#列, h[j]是最大能打高度.
        if h[j]>=h[i]:
            mat[i][k+1-j]=max(1+mat[i-1][k+1-(i)],mat[i-1][k+1-j])
        else:
            mat[i][k+1-j]=mat[i-1][k+1-j]
print(mat[-1][-1])

```

23421: 小偷背包

dp, <http://cs101.openjudge.cn/practice/23421>

思路：略。最简单的01背包。

代码：

```

def dp(n,b,v,w):
    dp=[[0]*(b+1) for i in range(n+1)]
    for i in range(1,n+1):
        for j in range(1,b+1):
            if j>=w[i-1]:
                dp[i][j]=max(dp[i-1][j],dp[i-1][j-w[i-1]]+v[i-1])
            else:
                dp[i][j]=dp[i-1][j]
    return dp[n][b]

```

n,b=map(int,input().split())

```
v=list(map(int,input().split()))
w=list(map(int,input().split()))
print(dp(n,b,v,w))
```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```
def dp(n,b,v,w):
    dp=[[0]*(b+1) for i in range(n+1)]
    for i in range(1,n+1):
        for j in range(1,b+1):
            if j>=w[i-1]:
                dp[i][j]=max(dp[i-1][j],dp[i-1][j-w[i-1]]+v[i-1])
            else:
                dp[i][j]=dp[i-1][j]
    return dp[n][b]

n,b=map(int,input().split())
v=list(map(int,input().split()))
w=list(map(int,input().split()))
print(dp(n,b,v,w))
```

02754: 八皇后

dfs and similar, <http://cs101.openjudge.cn/practice/02754>

思路：呜呜呜搜答案的，从第一行开始摆，摆到合理位置就往下摆，一行摆到头都不合理就返回上一行，八行摆完了就ans.append(A[:])然后退回最后一行继续往下摆

代码：

```
ans=[]
def queen(A, cur=0):
    if cur == len(A):
        ans.append(A[:])
        return 0
    for col in range(1, len(A)+1): # 遍历当前行的所有位置
        A[cur] = col
        for row in range(cur): # 检查当前位置是否相克
            if A[row] == col or abs(col - A[row]) == cur - row:
                break
        else: # 如果完成了整个遍历，则说明位置没有相克
            queen(A, cur+1) # 计算下一行的位置
queen([None]*8)
t=int(input())
for i in range(t):
    n=int(input())
    print(*ans[n-1], sep='')
```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```
ans=[]
def queen(A, cur=0):
    if cur == len(A):
        ans.append(A[:])
        return 0
    for col in range(1, len(A)+1): # 遍历当前行的所有位置
        A[cur] = col
        for row in range(cur): # 检查当前位置是否相克
            if A[row] == col or abs(col - A[row]) == cur - row:
                break
        else: # 如果完成了整个遍历, 则说明位置没有相克
            queen(A, cur+1) # 计算下一行的位置
queen([None]*8)
t=int(input())
for i in range(t):
    n=int(input())
    print(*ans[n-1], sep='')
```

189A. Cut Ribbon

brute force, dp 1300 <https://codeforces.com/problemset/problem/189/A>

思路: 用-1标记不可能分割成的长度, 剩下的等价于完全背包, 只不过每个物品的重量等于长度, 价值等于1

代码:

```
def dp(b,w):
    dp=[0]*4001
    for j in range(min(w),b+1):
        if j>=min(w):
            tmp=[0,0,0]
            for i in range(3):
                if (j-w[i]>=min(w) or j-w[i]==0) and dp[j-w[i]]!=-1:
                    tmp[i]=dp[j-w[i]]+1
            else:
                tmp[i]=-1
            dp[j]=max(tmp)

    return dp[b]
N,A,B,C=map(int,input().split())
print(dp(N,[A,B,C]))
```

代码运行截图 （至少包含有"Accepted"）

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
288744087	Practice: _12-_	189A - 39	PyPy 3-64	Accepted	124 ms	2108 KB	2024-10-29 16:56:14	2024-10-29 16:56:14		Compare

→ Source

Copy

```
def dp(b,w):
    dp=[0]*4001
    for j in range(min(w),b+1):
        if j>=min(w):
            tmp=[0,0,0]
            for i in range(3):
                if (j-w[i]>=min(w) or j-w[i]==0) and dp[j-w[i]]!=-1:
                    tmp[i]=dp[j-w[i]]+1
            else:
                tmp[i]=-1
            dp[j]=max(tmp)
    #print(dp[0:102])
    return dp[b]
N,A,B,C=map(int,input().split())
print(dp(N,[A,B,C]))
```

2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“计概2024fall每日选做”、CF、LeetCode、洛谷等网站题目。
OJ差两题，dp跟递归好难（哭）八皇后自己做的解法几百毫秒，上网搜了优化，开始懂什么叫有思路但代码写不出来了...